

ベクトルプロセッサの有効な使用法(1) : チューニングツールの使用法について

石氷, 結花
九州大学大型計算機センター研究開発部

河津, 秀利
九州大学大型計算機センター業務掛

菅崎, 直弘
九州大学大型計算機センター業務掛

山本, 美保子
九州大学大型計算機センター業務掛

他

<https://doi.org/10.15017/1468132>

出版情報 : 九州大学大型計算機センター広報. 18 (6), pp.552-579, 1985-11-25. 九州大学大型計算機センター
バージョン :
権利関係 :



ベクトルプロセッサの有効な使用法 (1)

— チューニングツールの使用法について —

石氷結花*, 河津秀利**, 菅崎直弘**, 山本美保子**, 武富 敬*

1. はじめに

文献 [1] で述べたように, Fortran プログラムをベクトルプロセッサ (以下 VP) で高速実行するためには, 可能な限り多くの演算を VP で実行可能なベクトル演算にする必要がある. Fortran 77/VP コンパイラは, プログラムを評価し, 可能であれば自動的にベクトル命令に翻訳する「自動ベクトル化」機能を持つので, とくに手を加えなくてもかなりの高速化が望めることがある. しかし, それで不十分な場合, あるいは, さらに効果を高めるためには, VP の特徴を生かすようにプログラムを改善 (チューニング) してやる必要がある. このためのチューニングツールとして, 以下のものがある.

i) 会話型 (バッチ型) ベクトライザ [2]

ii) FORTUNE [3]

iii) FORDAP77 [4]

これらのツールにより, チューニングの作業は以下のように進めるのがよい.

- 1) プログラムの動的解析により, 実行中枢部 (実行の集中している部分) を見出す. 通常, それらは数個のサブルーチンに集中していることが多い. この目的には, 上記 ii), iii) のツールが使用できる. また, 正確な計算時間分布の算出のためには, 付録 1 に示す方法で, タイマルーチンにより計測する方法もある (上記 iii) にはこの機能も含まれている).
- 2) 実行中枢部がベクトル化されているかどうかを調査する. このためには, 上記 i) のツールを使用する.
- 3) 上の 1), 2) で求めた情報をもとにして, 最適化制御行 (続稿 [5] 参照) の適用, プログラムの一部書き換えなどの作業を行う. この時にも, i) のツールが有用である.

しかし, 通常は, これら 1), 2) の情報をまとめて得るために, 最初にバッチ型ベクトライザを実行して見るのが一番早道である. これにより, 実行中枢部, その部分のベクトル化の可否, ベクトル化できない理由などの情報が直ちに得られる. ただし, 出力される情報が多いために, 大部のプログラムに対しては, 後述するようにベクトライザのオプションを指定して適度な出力に抑えることが望ましい. なお, プログラム高速化に関する

* 九州大学大型計算機センター 研究開発部

** 同 業務掛

続稿 [5] も併せて参照されたい。

2. 会話型およびバッチ型ベクトライザの使用法

2. 1 概要

会話型ベクトライザは、フルスクリーン型の端末で使用する（ただし、PFD-TTYの機能により、TTY端末での利用も可）。会話型ベクトライザは以下の機能を持つ。

- 1) FORTUNEモード・・・プログラムの動的解析を行う部分である。プログラム単位、DOループ単位、文単位の実行費用* を表示する。ただし、前もって、FORTUNEコマンド等で解析に必要なデータセットを用意しておく必要がある。ここで得られる情報は、次のベクトライズモードの動的効果表示の中に含まれるので、ここでは詳しくは述べない。
- 2) ベクトライズモード・・・FortranソースプログラムをFortran77/VPコンパイラで解析し、以下の情報を表示する。なお、このモードはバッチでも実行できる（バッチ型ベクトライザと呼ぶ）。
 - ・ベクトル化効果・・・各プログラムの実行回数、VPで実行した場合のおよその実行費用、スカラ（汎用計算機）で実行した場合のおよその実行費用、全演算数のうちどれだけベクトル化できるかを示したベクトル化率とそれらを総合して期待できるおよその効果（倍率）の出力。これらの情報の詳細については、実際の出力例とともに2. 2. 1で述べる。ベクトル化効果の表示には以下の3種がある。
 - i) 動的効果表示・・・実際にプログラムを実行して得られた実行解析情報（FORTUNEによる）をもとに、ベクトル化情報を出力する
 - ii) 静的効果表示・・・プログラムの内容から予想されるDOループの回転数（以後「ベクトル長」という）やIF文の真になる確率を仮定してベクトル化情報を出力する
 - iii) 単位効果表示・・・"静的効果表示"をプログラム単位内に限定してベクトル化情報を出力する
 - ・ベクトル化情報・・・ベクトル化表示付きソースプログラムリスト（V、M、Sなどの記号で、実行文のベクトル化の度合を示す）。それぞれのDOループについてどのDO変数でベクトル化されたか、あるいは、こういう理由でベクトル化されないといったベクトル化メッセージの出力。
 - ・チューニング情報・・・こういう具合に修正すればベクトル化可能であるというメッセージの出力。特に、会話型に限れば、例題付の詳細なアドバイスの表示、そのメッセージを見ながらソースプログラムも表示できる画面分割機能。さらに、ベクトライザからの質問文に答えるだけで容易にプログラムの修正・再実行

* 実行費用とは、単純な代入文を1回実行するのに必要な費用を1とした場合の相対的な値であり、ある実行文を実行するのに必要な時間の目安とすることができる。

行が行えるコマンドなどチューニング機能が充実している。

ベクトライザからのチューニングメッセージに従ってプログラムを修正し、即時に再実行してベクトル化情報を確認するには、会話型ベクトライザは手軽であるが、一般にVPで実行したいプログラムは、ステップ数が多く実行時間もかなりかかるものであるため、いきなり会話型で実行すると、結果が表示されるまで長時間端末の前で待たされることになりかねない。そこで、まずはじめに、バッチ型ベクトライザで実行し、プログラム全体にわたる実行解析情報、ベクトル化情報、ベクトル化／チューニングメッセージをまとめて入手して、実行の中心となっている箇所はどこか、チューニングなしでどの程度自動ベクトル化されているか、どの部分がベクトル化不能かなどの情報を得る。それらを把握した上で、どのサブルーチン、あるいはどのDOループに改善の余地があるかを検討し、必要であれば会話型ベクトライザにかける方法を推奨したい。

また、この実行はあくまでもテストランであるから、必要最小限度の時間（せいぜいCPU時間2～3分程度）で切り上げるのがよい。

2. 2 使用法と出力情報の見方

2. 2. 1 バッチ型ベクトライザ

バッチ型ベクトライザの実行には、カタログドプロシジャINTVCTを使用する。

プロシジャ名	記 号 パ ラ メ ー タ
INTVCT	DSN='ソースデータセット名' [, SYSOUT= { <u>A</u> K S O U}] [, FEATURE= { <u>D</u> S U}] [, STEP= { <u>CGO</u> CLG}] [, PRVLIB='データセット名']

・記号パラメータの説明

DSN：解析したいソースデータセット名を指定する。必須パラメータである。

SYSOUT：出力クラスを指定する。A（PL／I 60字セットLP）、K（カナ付LP）、S（英小文字付LP）、O（日本語LP）、U（オープンNLP）のいずれか。省略値はA。

FEATURE：ベクトル化効果表示の種別を指定する。省略値はD。

D 動的効果表示 (2 または 4 プロシジャステップ, FORTUNE の実行あり)
 S 静的効果表示 (1 プロシジャステップ, FORTUNE の実行なし)
 U 単位効果表示 (1 プロシジャステップ, FORTUNE の実行なし)

STEP: 処理過程を選択する。省略値は CGO。

CGO FORTUNE を 1 プロシジャステップで実行する
 CLG FORTUNE を 3 プロシジャステップで実行する

PRVLIB: 組み込みたい私用ライブラリのデータセット名を指定する。

・注意事項

i) FORTUNE のオプションを指定したい時 (FEATURE=D の時) は, 以下のように指定する。

PARAM.FTUN='FORTUNE のオプション/Portran77 コンパイラオプション'

ii) ベクトライザのオプションを指定したい時は, 以下のように指定する。この時指定可能なベクトライザのオプションを表 2. 1 に, Fortran77/VP コンパイラオプションを表 2. 2 に掲げる。

PARAM.INTVCT='ベクトライザのオプション/Portran77/VP コンパイラオプション'

表 2. 1 パッチ型ベクトライザで指定できるベクトライズオプション一覧表

オプションの形式	機能概要
$\left\{ \begin{array}{l} \text{DYNAMIC} \\ \text{D} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{STATIC} \\ \text{S} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{UNIT} \\ \text{U} \end{array} \right\}$	ベクトライズ情報の表示種別を指定する。 動的効果表示を求める (会話型ベクトライザの動的効果表示 (1) に対応する) 静的効果表示を求める 単位効果表示を求める
(RNAME (* name1, name2, ...name10))	ベクトライズ情報を出力するルーチン名を指定する, (ルーチン名は最大 10 個まで指定できる)。* が指定された場合は, プログラム中の全ルーチンの情報を出力する。
SEPARATE NOSEPARATE	文単位ベクトライズ情報/ループ一覧/メッセージ一覧の各情報をページ替えて出力するか否かを指定する。
STATEMENT NOSTATEMENT	文単位ベクトライズ情報を出力するか否かを指定する。NOSTATEMENT オプションが指定された場合には, プログラム単位ベクトライズ情報のみが出力される。
PUNIT NOPUNIT	ルーチンのプログラム単位ベクトライズ情報を文単位ベクトライズ情報の直後に出力するか否かを指定する。

表2. 2 バッチ型ベクトライザで指定できるFortran77/VPオプション一覧表

オプションの形式		機能概要
<u>ADVANCED</u> ({ <u>EVL</u> NOEVL })	<u>NOADVANCED</u> ({ EVL <u>NOEVL</u> })	実行モードの選択
<u>ADV</u> ({ <u>EVL</u> NOEVL })	<u>NOADV</u> ({ EVL <u>NOEVL</u> })	(簡略形)
<u>AE</u>	<u>NOAE</u>	拡張リージョンの割付け
<u>ALIGNC</u>	<u>NOALIGNC</u>	共通ブロック内のデータに対する 記憶領域の割付け方法
<u>BYNAME</u>	<u>NOBYNAME</u>	仮引数と実引数の結合方法
<u>DEBUG</u> [(dpara)]	<u>NODEBUG</u>	デバッグ機能
<u>DOUBLE</u>		自動精度拡張機能
<u>QUAD</u>		自動精度拡張機能
<u>SCALAR</u>		ベクトル化の抑止
<u>SIZE</u> ({ MAX nK })		コンパイラの使用する仮想記憶領域の 大きさ
<u>VOCL</u>	<u>NOVOCL</u>	最適化制御行の有効/無効
<u>VP</u> [({ ANY 100 200 })]		FORTRAN77/VPコンパイラの呼出し
<u>VMSG</u> ({ L DETAIL })	<u>NOVMSG</u>	ベクトル化メッセージを出力するか 否かの指定
<u>XOPT</u> (xpara)		オブジェクトモジュールの最適化 機能の制御

備考1. dparaは、ARGCHK, SUBTRACE, TRACE, INIT, SUBCHK, 1OVERFL, 及びUNDEFの並びである。

備考2. nは、24～16000の数字である。

備考3. xparaは、{ IL | INIL | NOIL }, { AMOVE | NOAMOVE } 及び { MSG | NOMSG } の並びである。

備考4. VMSGは、バッチ型ベクトライザでのみ指定が有効である。

以下に、バッチ型ベクトライザの実行を行うJCLの例を3つ示す。プログラムのステップ数がそれほど長くないものであれば、①の方法で全ベクトル化情報を出力する事が一番早道だが、サブルーチンの数が多く、ステップ数の長いプログラムなどは、まず②の方法で実行中枢部のサブルーチンを見つけ出し、それから③のジョブで、該当するサブルーチンのみのベクトル化情報を収集する方法が効率的である。利用者各々のプログラムの大きさや、必要とする情報に応じて試みられたい。

- ① 全ベクトル化情報 (図1. 1～1. 3) を出力する。コンパイラオプションとしてOPT(3)などを指定している。

```
//F9999A JOB パスワード,CLASS=A,.....
//      EXEC  INTVCT,DSN='ソースプログラム名',
//      PARM.FTUN='COMPILE/GO,NOPRINT,OPT(3)',
//FTUN.SYSGO DD  *
//      ( データ )
//
```

② ベクトライザオプションのNOSTATEMENTを指定して、プログラム単位ベクトライズ情報(図1.

3)のみを出力する.

```
//F9999A JOB パスワード,CLASS=A,.....
//      EXEC  INTVCT,DSN='ソースプログラム名',
//      PARM.FTUN='COMPILE/GO,NOPRINT,OPT(3)',
//      PARM.INTVCT='D,NOSTATEMENT'
//FTUN.SYSGO DD  *
//      ( データ )
//
```

③ ②で得られた情報をもとにして、実行中サブルーチンに対するベクトル化情報を得る.

```
//F9999A JOB パスワード,CLASS=A,.....
//      EXEC  INTVCT,DSN='ソースプログラム名',
//      PARM.FTUN='COMPILE/GO,NOPRINT,OPT(3)',
//      PARM.INTVCT='D,RNAME(サブルーチン名,.....)'
//FTUN.SYSGO DD  *
//      ( データ )
//
```

図1. 1～1. 3に①のJCLを実行した時の出力を示す。出力順としては、各プログラム単位別に、文単位ベクトライズ情報(図1. 1), DOループ一覧およびメッセージ一覧情報(図1. 2)が続き、最後にプログラム単位ベクトライズ情報(図1. 3)となっている。まず図1. 1であるが、ソースプログラムの左にV-COST(後述)と、その文がベクトル化されるかどうかのV,M,Sの記号で示される。Vであればベクトル化可能、Mでは一部ベクトル化可能、Sはベクトル化不可能なためスカラで実行されることを意味する。ソースプログラムの右側には、S-COST(後述)、実行回数、IF文の条件式が真になる割合が出力される。図1. 2には、このプログラム単位の中にあらわれるDOループの行番号と、それぞれがどれだけの実行費用を持つか、またベクトル長(DOの回転数)はいくらであるかが出力されている。また、そのひとつひとつのDOループについてベクトル化/チューニングメッセージが付加されている。

図1. 3のベクトル化情報の内容は、以下のようなものである。(出力例の左から順に簡単に解説する)。

INTERACTIVE VECTORIZER		V10L20	DATE 85.09.12 TIME 07.18.09				PAGE	3
LINE-NO.	V-COST	C----	STATEMENT (MAIN)	-----3-----4-----5-----6-----7--	S-COST	EXECUTIONS	TRUE	
00001050	10333 V		VC(I,J)=UC(I,J)*YD(I,K)		620000S	620000E		
00001060	248000 V		VEN=SQRT(UC(N,J)**2+VC(N,J)**2)		1488000S	620000E		
00001070	10333 V		UC(N,J)=VEN*CS(N,K)		620000S	620000E		
00001080	31666 V	100	VC(N,J)=UC(N,J)*YD(N,K)		190000S	620000E		
00001090		C			S	E		
00001100		C	--- INTERPOLATE TO KNOT POINT ---		S	E		
00001110	4000		CALL CRVFIT(UC,U,CU,INEX)		40000S	2000E		
00001120	4000		CALL CRVFIT(VC,V,CV,INEX)		40000S	2000E		
00001130		C			S	E		
00001140		C	--- UPSTREAM BOUNDARY CONDITION ---		S	E		
00001150	2000		SP=0.0		20000S	2000E		
00001160	10000 V		DO 110 I=1,N		100000S	2000E		
00001170	27333 V	110	SP=SP+P(I,1)		1640000S	800000E		
00001180	16000		SP=SP/N		160000S	2000E		
00001190	2000		AP=SP*7		20000S	2000E		
00001200	112000		SD=AP**0.7142857		1120000S	2000E		
00001210	58000		VE=SQRT(1-AP/SD)		580000S	2000E		
00001220	4000		VU=VE*CS1		40000S	2000E		
00001230	2000		VU=VU*YD1		20000S	2000E		
00001240	18000 V		DO 115 I=1,N		180000S	2000E		
00001250	13333 V		P(I,1)=SP		800000S	800000E		
00001260	13333 V		D(I,1)=SD		800000S	800000E		
00001270	13333 V		U(I,1)=VU		800000S	800000E		
00001280	40333 V	115	V(I,1)=VU		2420000S	800000E		
00001290		C			S	E		
00001300		C	--- OUTPUT PROG. -----		S	E		
00001310	4001		*VOCL STMT,IF(0)		6001S	2000E	0.0%	
00001320	19990		IF(INEX.EQ.IEND) GO TO 150		19990S	1999E		
			IP=MOD(INEX,LIST)		S	E		
			*VOCL STMT,IF(99)		S	E		
00001330	5978		IF(IP.NE.0) GO TO 130		5978S	1999E	99.0%	
00001340	38		CALL OUTPUT(RATIO,INEX)		38S	19E		
00001350	190		IP=MOD(INEX,IPLT)		190S	19E		
			*VOCL STMT,IF(84)		S	E		
00001360	54		IF(IP.NE.0) GO TO 130		54S	19E	84.2%	
00001370		CF	CALL AXYPL(P,INEX,RATIO)		S	E		
00001380			XO=15		S	E		
00001390			YO=0.0		S	E		
00001400		CF	CALL PLOT(XO,YO,-3)		S	E		
00001410	12		GO TO 130		12S	3E		
00001420	2	150	CALL OUTPUT(RATIO,INEX)		2S	1E		
00001430	2		CALL FLCON		2S	1E		
00001440	2		CALL LISTUP		2S	1E		
			*VOCL LOOP V1(J)		S	E		
00001450	6 S		DO 140 J=1,M		6S	1E		
00001460	500 V		DO 140 I=1,M		500S	100E		
00001470	24783 V	140	ZK(I,J)=SQRT((U(I,J)**2+V(I,J)**2)*D(I,J)/1.4/P(I,J))		148202S	4000E		
00001480		CF	CALL AXYPL(P,INEX,RATIO)		S	E		
00001490			AMA=P(1,1)		S	E		
00001500			AMA=P(1,1)		S	E		
00001510	6 S		DO 200 I=1,M		6S	1E		
00001520	200 V		DO 200 J=1,M		200S	40E		
			*VOCL STMT,IF(1)		S	E		
00001530	1004 V		IF(P(I,J).GT.AMA) AMA=P(I,J)		12050S	4000E	1.2%	

図1. 1 文単位ベクトライズ情報

ベクトルプロセッサの有効な使用法(1) — チューニングツールの使用法について —

VECTORIZE - ROUTINE INFORMATION (DYNAMIC FEATURE)							PAGE	
ROUTINE FLAG	MSG	EXECUTIONS	V-COST	S-COST	V-RATE	V-EFFECT		
MAIN	I	8	1	34526873	476844852	98.8	5.6-10.0	1

VECTORIZE - LOOP LIST --- MAIN				PAGE	
LINE-NUMBER	V-COST	S-COST	V-LENG		
00000480-00000480	18513333	111080000	40		
00000750-00000780	7273333	87280000	99		
00000850-00000850	5280166	43362000	99		
00000980-00000980	5280166	63362000	99		
00000360-00000380	4626666	55520000	99		
00000890-00000910	4626666	55520000	99		
00000640-00000640	1986666	23840000	99		
00000470-00000480	1188000	11880000	99		
00000880-00000920	964000	9640000	40		
00000740-00000790	962000	9620000	40		
00000350-00000390	882000	8820000	40		
00001030-00001080	568998	3414000	31		
00000580-00000580	354471	6635662	97		
00000320-00001410	535582	535582	2000		
00000630-00000640	440333	6420000	40		
00000840-00000850	320000	3200000	40		
00000970-00000980	320000	3200000	40		
00001250-00001280	80332	4820000	40		
00000550-00000580	63840	143640	40		
00001170-00001170	27333	1640000	40		
00001470-00001470	24683	1481022	40		
00000190-00000220	3347	20096	40		
00001330-00001340	1474	20096	100		
00001640-00001650	1671	20067	100		
00000180-00000220	600	600	100		
00001460-00001470	600	600	100		
00001520-00001540	200	200	40		
00001630-00001650	160	160	40		

VECTORIZE - MESSAGE LIST --- MAIN	
JZK230I-I	THE PROGRAM UNIT HAS SEQUENCE ERROR.
JND102I-I	LWD:00000170
	THE 1ST OPTIMIZATION CONTROL LINE OF THIS STATEMENT HAS INVALID SYNTAX IN THE 1ST PARAMETER.
JND211I-I	LWD:00000320
	DO WHILE, DO UNTIL, OR IF/GOTO LOOP IS NOT VECTORIZABLE.
JND273I-I	LWD:00000340
	PARTIAL VECTORIZATION OVERHEAD IS TOO LARGE.
JND102I-I	LWD:00000460
	THE 1ST OPTIMIZATION CONTROL LINE OF THIS STATEMENT HAS INVALID SYNTAX IN THE 1ST PARAMETER.
JND273I-I	LWD:00000730
	PARTIAL VECTORIZATION OVERHEAD IS TOO LARGE.
JND273I-I	LWD:00000870
	PARTIAL VECTORIZATION OVERHEAD IS TOO LARGE.
JND102I-I	LWD:00001450
	THE 1ST OPTIMIZATION CONTROL LINE OF THIS STATEMENT HAS INVALID SYNTAX IN THE 1ST PARAMETER.

図1. 2 DOループ一覧およびメッセージ一覧情報

INTERACTIVE VECTORIZER		V10L20		DATE 85.09.12		TIME 07.18.09		PAGE		37
VECTORIZE - TOTAL INFORMATION (DYNAMIC FEATURE)										
ROUTINE	FLAG	MSG	EXECUTIONS	V-COST	S-COST	V-RATE	V-EFFECT	PAGE		
(TOTAL)	I	101	-----	208405712	1335167355	97.9	4.3- 7.4			

SMOOTH	I	3	8000	88273331	495720000	98.5	3.8- 6.4	15		
MAIN	I	8	1	34526873	476844852	98.8	5.6-10.0	1		
NEUSTP	I	1	6000	50049750	262534000	97.1	3.6- 5.9	11		
CRVFIT	I	1	6000	8853121	93027612	98.7	6.4-11.9	14		
OUTPUT	OK	8	20	6447089	6635760	3.4	1.0- 1.0	17		
FLCDM	OK	5	1	198028	202987	2.9	1.0- 1.0	20		
ORTH	OK	10	1	26860	75993	77.5	2.3- 2.9	7		
INICOM	I	5	1	16911	78469	94.1	3.3- 5.1	9		
LISTUP	OK	0	1	10085	10243	1.8	1.0- 1.0	19		
SCTR	OK	1	1	3664	17439	94.6	3.4- 5.3	27		
BLOCKD	OK	0	0	0	0	0.0	0.0- 0.0	6		
PCDUM	OK	14	0	0	0	0.0	0.0- 0.0	22		
FPLT	OK	0	0	0	0	0.0	0.0- 0.0	26		
AXYPL	OK	23	0	0	0	0.0	0.0- 0.0	28		
MCDUM	OK	14	0	0	0	0.0	0.0- 0.0	31		
STREAM	OK	8	0	0	0	0.0	0.0- 0.0	35		

図1. 3 プログラム単位ベクトライズ情報

項目名	内 容
ROUTINE	プログラム単位名がV-COSTの高い順に出力される。
FLAG	Fortran 77/VPコンパイラの診断メッセージのエラーレベルを示す。OKであれば、診断メッセージなし。I, W, E, S, Uは、通常のFortran 77コンパイラの診断メッセージに同じ。
MSG	Fortran 77/VPコンパイラから出力された診断/ベクトル化/チューニングメッセージ数の合計。
EXECUTIONS	各プログラム単位の実行回数。ただし、"動的効果表示"を指定した場合には、実行解析データセットから求めた実際の呼び出し回数が出力されるが、"静的効果表示"の場合はベクトライザが仮定した値となる。"単位効果表示"であれば、各プログラム単位の呼び出し回数はすべて1となる。
V-COST	各プログラム単位をVPで実行した時の、およその実行費用を表わす。(TOTAL)の行には、全プログラムの総和が出力される。
S-COST	各プログラム単位をスカラ(汎用計算機)で実行した時の、およその実行費用を表わす。(TOTAL)の行には、全プログラムの総和が出力される。
V-RATE	各プログラムのおよそのベクトル化率を示す。
V-EFFECT	ベクトル長(DOループの回転数)、データの利用関係、演算子の種類、演算子の数などから算出したおよそのベクトル化効果(倍率)を上限から下限の範囲で出力する。

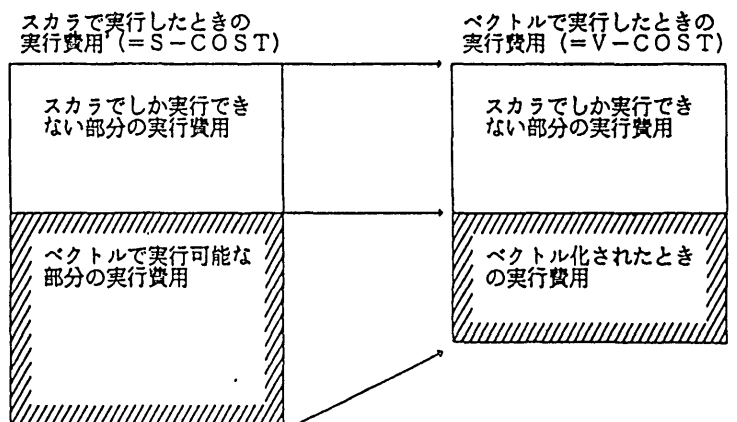
これらのうち、V-COST、S-COST、V-RATE、V-EFFECTの関係を算出式を図1. 4に掲げる。

なお、ベクトライザの目的はVP上での実際の実行の効果を見積もるのであるから、ベクトライザの実行はあくまでM380(汎用機)上でのテストランでしかない。よって、実際にVPで走らせる時には、ベクトライザで得られた情報と異なるベクトル化が行われることもある。一例をあげると、多重DOループの場合では、ベクトライザは最も内側の制御変数に対してベクトル化の解析を行うが、FORTRAN 77/VPコンパイラは、最内側でなくても最適なDO変数を選んでベクトル化を行う。ベクトライザの出力情報は材料であって、結果ではないことに注意されたい。

2. 2. 2 会話型ベクトライザ

会話型ベクトライザを使用する場合は、TSSのREADY状態でコマンド INTVCT を投入する。

会話型ベクトライザは、PFDのもとで動作するので、メニュー画面で処理番号を選択したり、パラメタの入力だけで容易に処理を進行させることが出来る。ここでは、代表的なパネルの遷移にそって、バッチの例と同じ"動的効果表示"を選択してベクトル化情報を収集し、さらに、画面分割機能を用いてソースプログラムの



$$V-RATE : \frac{\text{ベクトルで実行可能な部分の実行費用}}{\text{スカラで実行したときの実行費用 (= S-COST)}} \%$$

$$V-EFFECT : \frac{\text{スカラで実行したときの実行費用 (= S-COST)}}{\text{ベクトルで実行したときの実行費用 (= V-COST)}}$$

図1. 4 V-COST, S-COST, V-RATE, V-EFFECTの関係

書き換えを行う例を紹介する。

会話型ベクトライザで利用できるコマンドを以下に示す。* を付したコマンドは、PFDの同名のコマンドとは機能が異なるため、注意を要する。

表 2. 3 会話型ベクトライザで利用できるコマンド一覧

コマンド名	機 能
SELECT	メンバ名などの一覧表が表示されている時に、その中の項目を選択する。
LOCATE	メンバ名などの一覧表が表示されている時に、指定した項目の箇所に画面を位置づける。
MAX	スクロールキーと併用して、指定されているデータセットの先頭（あるいは最後）の情報を、表示画面の先頭（最後）に表示する。
SAVE *	文単位ベクトライズ情報を修正／編集した結果を、元のデータセットに保存する。ただし、ENDキー（PF3）で画面分割での文単位ベクトライズ情報の修正／編集を完了してからでないと使用できない。また、複数あるいは全てのメンバを選択した時には、SAVE処理は行われない。
CANCEL *	プログラム単位ベクトライズ情報画面、ループ一覧画面、およびメッセージ一覧画面で有効であり、修正／編集したプログラムを元のデータセットに保存せずに、表示中の画面の処理を終了させる。ただし、画面分割中にこのコマンドは利かない。
DLOOP	プログラム実行解析情報のうち、実行費用の高いDOLoopの一覧表や、実行費用の最も高いDOLoopを表示する
NEXT	DOLoop表示画面において、次に実行費用の高いDOLoopを表示する
LIST	現在表示中の情報をSYSOUTへ出力する
VCHECK	文単位ベクトライズ情報を修正／編集したルーチンに対して、新たにベクトル化情報を収集する。ただし、ベクトル化効果は単位効果表示となる。
REPLY	チューニングメッセージとしての質問文に対する応答により、最適化制御行を作成する。

i) 起動と終了

(1) 起動

端末が 'READY' の状態にあるとき、次のコマンドを入力する。

INTVCT

このコマンドの入力により、プライマリオプションメニュー (図2. 1) が表示される。

(2) 終了

プライマリオプションメニューでEND (PF3) キー、またはRETURN (PF4) キーを押下するか、またはオプションの選択コードとして 'X' を選択する。

ii) 使用法

FORTUNEの解析情報データセットを用いてベクトル化情報を表示する '動的効果表示(1)' の使用法についてパネルを中心に説明する。

(1) 表示情報の選択

・プライマリオプションメニュー (図2. 1)

コマンド 'INTVCT' を入力すると会話型ベクトライザが起動され、プライマリオプションメニューが表示される。

番号/コード	オプション名	機 能
1	VECTORIZE	FORTRAN 77/VPコンパイラを用いて原始プログラムの解析を行い、ベクトル化情報を表示する。
2	FORTUNE	FORTUNEシステムを動作させてすでに得られている原始プログラムのプログラム実行解析情報を表示する。(3. 2参照)
X	EXIT	会話型ベクトライザを終了する。

ここでは、1 (VECTORIZE) を選択する。

```

----- INTERACTIVE VECTORIZER PRIMARY OPTION MENU V10L20 -----
SELECT OPTION ==> 1_                                USERID - F0022
-----
 1 VECTORIZE - DISPLAY THE VECTORIZED INFORMATION
 2 FORTUNE   - DISPLAY THE INFORMATION WHICH IS ANALYZED BY FORTUNE
 X EXIT     - TERMINATE VECTORIZER
-----

PRESS END KEY TO TERMINATE VECTORIZER

```

図2. 1 プライマリオプションメニュー

(2) 入力データセットの指定

- ・ソースデータセット種別選択メニュー (図2. 2)

ベクトル化しようとする原始プログラムの入ったデータセットの種類を指定する。

1 : 一般のデータセット (順, 区分データセットの場合)

・ 2 : GEMライブラリデータセット

ここでは, 1 (一般のデータセット) を選択する。

```

----- VECTORIZE - DATASET SELECTION MENU -----
SELECT OPTION ==> 1_
 1 GENERAL LIBRARY - SOURCE DATASET IN THE GENERAL LIBRARY
 2 GEM LIBRARY    - SOURCE DATASET IN THE GEM LIBRARY

```

図2. 2 ソースデータセット種別選択メニュー

- ・ソースデータセット指定メニュー (図2. 3)

ソースデータセット名を指定する。ここでは, ' F 0 0 2 2. VPEXAM. FORT77 'を指定している。

```

----- VECTORIZE - SOURCE DATASET MENU -----
SPECIFY PARAMETERS BELOW:

LIBRARY DATASET:
PROJECT ==> F0022
LIBRARY ==> VPEXAM
TYPE    ==> FORT77_
MEMBER  ==>          (BLANK TO DISPLAY MEMBER LIST)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>          (IF NOT IN CATALOG)

DATASET PASSWORD ==>          (IF PASSWORD PROTECTED)

```

図2. 3 ソースデータセット指定メニュー

・メニュー選択メニュー (図2. 4)

インクルードデータセット [2] の指定メニューを表示するかどうかを指定する。ここでは、表示する必要がないので 'NO' を指定している。

----- VECTORIZE - MENU SELECTION MENU -----
 ENTER/VERIFY PARAMETER BELOW:

MENU NAME TO BE DISPLAYED
 INCLUDE DATASET MENU ==> NO (YES/NO)

図2. 4 メニュー選択メニュー

・(3) 表示種別の選択

・表示種別選択メニュー (図2. 5)

プログラムをベクトルプロセッサで実行したときに得られるベクトル化効果の表示を選択する。

番号	オプション名	機 能
1	動的効果表示(1)	すでに求めてあるプログラムの実行解析結果のデータセット (FORTUNEの解析情報データセット) を用いてベクトル化情報を表示する。
2	動的効果表示(2)	解析対象プログラムに対して、FORTUNEを呼出しプログラムを実行させてプログラムの実行解析結果の情報を得た後、そのデータセットを用いてベクトル化情報を表示する。
3	静的効果表示	解析対象プログラムに対して、ループの回転数、IF文のTRUE率および各ルーチンのプログラム中での呼出し回数を考慮してベクトル化情報を表示する。
4	単位効果表示	解析対象プログラムに対して、ループの回転数及びIF文のTRUE率を仮定して、ルーチン単位内に閉じたベクトル化情報を表示する。

ここで、1 (動的効果表示(1)) を選択する。

----- VECTORIZE - FEATURE SELECTION MENU -----

SELECT OPTION ==> 1_

- 1 DYNAMIC FEATURE(1) - CALCULATE THE COST USING FORTUNE ANALYZED DATA SET
- 2 DYNAMIC FEATURE(2) - CALL FORTUNE AUTOMATICALLY AND CALCULATE THE COST USING FORTUNE ANALYZED DATA SET
- 3 STATIC FEATURE - CALCULATE THE COST USING THE ASSUMED VALUE
- 4 UNIT FEATURE - CALCULATE THE COST OF EACH PROGRAM UNIT USING THE ASSUMED VALUE

図2. 5 表示種別選択メニュー

- ・ FORTUNE解析情報データセット指定メニュー (図2. 6)

表示種別選択メニューで ' 1 ' を選択すると, FORTUNE解析情報データセット指定メニューが表示される. FORTUNE解析情報データセットは, 動的効果表示(1)を求めるために必要なデータセットで, ベクトライズモードの入力となる原始プログラムと同一のプログラムをもとにFORTUNEを使用して事前に作成しておかなければならない.

```
----- VECTORIZE - FORTUNE ANALYZED DATASET MENU -----
SPECIFY PARAMETERS BELOW:

LIBRARY DATASET:
PROJECT ==> F0022
LIBRARY ==> VPEXAM
TYPE    ==> LIST_

OTHER SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>          (IF NOT IN CATALOG)

DATASET PASSWORD ==>          (IF PASSWORD PROTECTED)
```

図2. 6 FORTUNE解析情報データセット指定メニュー

(4) ベクトライズオプションの指定

- ・ ベクトライズオプションメニュー (図2. 7)

ベクトライズモードでは,

- ・ FORTRAN77/VPコンパイラ
- ・ FORTUNEコンパイラ
- ・ FORTRAN77コンパイラ
- ・ ロード
- ・ 実行時

のオプションが指定でき, このメニュー画面で指定する. また, 各オプション入力フィールドに複数のオプションを指定する場合には, オプションとオプションの区切りは ', ' でなければならない.

必要なオプションを指定した後ENTERキーを押下すると, FORTRAN77/VPコンパイラが呼出され, 解析が始まる.

```

----- VECTORIZE - OPTION MENU -----
SELECT DESIRED OPTIONS:

FIXED      I FREE      ==> 1  (1-FIXED      2-FREE      )
EBCDIC     I BCD       ==> 1  (1-EBCDIC     2-BCD       )
ASTER      I NOASTER   ==> 2  (1-ASTER     2-NOASTER    )
LMSG       I SMSG      ==> 1  (1-LMSG      2-SMSG      )
LANGVL(66) ==> 2  (1-LANGLVL(66) 2-LANGLVL(77))
FLAG(IIWIEIS) ==> 1  (I W E S)

SPECIFY OTHER OPTIONS:

FOR FORTRAN 77/VP ==> ADVANCED

(IF DYNAMIC FEATURE(2) SELECTED)
FOR TORTUNE      ==>
FOR FORTRAN 77   ==>
FOR LOADER       ==>
FOR RUN-TIME     ==>

```

図2. 7 ベクトライズオプションメニュー

(5) ベクトライズ情報表示画面

ベクトライズ情報表示画面には、以下のものがある。

- ・プログラム単位ベクトライズ情報画面 (図2. 8)
- ・ループ一覧画面 (図2. 9)
- ・メッセージ一覧画面 (図2. 10)
- ・詳細情報画面 (図2. 11, 図2. 12)

・プログラム単位ベクトライズ情報画面 (図2. 8)

この画面で表示されている情報の内容は、バッチ型ベクトライザの場合と同じである。

この画面でルーチン名の左側に、行サブコマンド ' S 'を入力するか、SELECTコマンドによ

りルーチン名を指定すると、図2. 9に示すルーチン内のループ一覧が表示される。

```

VECTORIZE - TOTAL INFORMATION (DYNAMIC FEATURE) ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

ROUTINE  FLAG  MSG  EXECUTIONS      V-COST      S-COST      V-RATE      V-EFFECT
(TOTAL)  OK    94      -----      234702462  1467971355  97.7      4.2- 7.1

SMOOTH    OK      0      8000      88273331  495720000  98.5      3.8- 6.4
NEWSTP    OK      4      6000      76346500  395358000  96.7      3.6- 5.8
MAIN      OK      4      1      54526873  476844852  98.8      5.6-10.0
S CRVFIT  OK      3      6000      8853121  93027612  98.7      6.4-11.9
-OUTPUT   OK      8      20      6447089  6635760   3.4      1.0- 1.0
FLCON     OK      5      1      198028   202987    2.9      1.0- 1.0
ORTH      OK     10      1      26860    75993    77.5      2.3- 2.9
INICON    OK      3      1      16911    78469    94.1      3.3- 5.1
LISTUP    OK      0      1      10085    10243    1.8      1.0- 1.0
SCTR      OK      1      1      3664     17439    94.6      3.4- 5.3
BLOCKD    OK      0      0      0         0         0.0      0.0- 0.0
*** 終り ***

```

図2. 8 プログラム単位ベクトライズ情報画面

・ループ一覧画面 (図 2. 9)

この画面で表示されている情報の内容を以下に示す。

項 目	内 容
LINE-NUMBER	D O ループ, あるいは I F ループの開始行番号と終了行番号が表示される。
V-COST	ループ内のベクトルコストの合計値が表示される。
S-COST	ループ内のスカラーコストの合計値が表示される。
V-LENG	ループの平均実行回数 (ベクトル長) が表示される。

この画面で, ループを選択して E N T E R キーを押下すると, メッセージ一覧画面 (図 2. 1 0) が表示される。

```

VECTORIZE - LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600 5920000    71040000    99
S 00003530-00003530 2983950    35807400    97
-00003580-00003600 520000     720000     40
00003510-00003530 179550     307800     40
*** 終り ***
  
```

図 2. 9 ループ一覧画面

・メッセージ一覧画面 (図 2. 1 0)

ループ一覧画面で選択されたループに関するベクトライズ/チューニングメッセージを先頭に位置付けて, メッセージ一覧が表示される。診断メッセージのメッセージ一覧画面を表示させるには, ベクトライズ/チューニングメッセージのメッセージ一覧が表示された時点で, ' M A X ' コマンドと U P (P F 7) キーを押下することにより表示することができる。

この画面でメッセージ I D を選択して E N T E R キーを押下すると, 詳細情報画面 (図 2. 1 1) が表示される。

```

VECTORIZE - MESSAGE LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

S_JND2011-I LNO:00003510 - 00003520
THE STATEMENTS IN THIS RANGE ARE VECTORIZED BY DO VARIABLE I.
JND2011-I LNO:00003530
THE STATEMENTS IN THIS RANGE ARE VECTORIZED BY DO VARIABLE J.
JND2011-I LNO:00003600
THE STATEMENTS IN THIS RANGE ARE VECTORIZED BY DO VARIABLE J.
*** 終り ***

```

図 2. 1 0 メッセージ一覧画面

・詳細情報画面 (図 2. 1 1, 図 2. 1 2)

メッセージ一覧画面で選択されたメッセージ ID に対する詳細な情報が表示される。ここでは、診断メッセージに対応する方法や、選択したベクトライズ/質問文に対応するベクトル化されたプログラムの例などが表示される。

```

----- VECTORIZE DETAIL INFORMATION - JND2011 -----(1/1)
SELECT ==> 1_
GENERAL INFORMATION

THERE ARE THREE IDEAS FOR GETTING MORE EFFICIENT OBJECT MODULES.

1. PERFORMANCE IMPROVEMENT BY INCREASING THE VECTOR LENGTH
2. PERFORMANCE IMPROVEMENT BY INCREASING THE NUMBER OF VECTOR OPERATIONS
3. PERFORMANCE IMPROVEMENT BY MEMORY ACCESS OPTIMIZATION

PLEASE SELECT THE APPROPRIATE NUMBER FOR DETAILS ON EACH METHOD.

```

図 2. 1 1 詳細情報画面(1)--JND2011--

```

----- VECTORIZE DETAIL INFORMATION - JND2011 -----(1/1)
1. PERFORMANCE IMPROVEMENT BY INCREASING THE VECTOR LENGTH

- VECTOR OPERATION WITH LONG VECTOR IS EXECUTED MORE EFFICIENTLY THAN
  THAT WITH SHORT VECTOR.
  ONE METHOD FOR INCREASING VECTOR LENGTH IS SHOWN IN THE FOLLOWING :

LOOP DISTRIBUTION AND INDEX EXCHANGE

+---> DO 10 I=1,100
|      X =
| +--> DO 10 J=1,10
| |
| |      = X
| |
| + 10 CONTINUE
|
+---> DO 11 I=1,100
|      WX(I) =
| +-- 11 CONTINUE
| +---> DO 10 J=1,10
| + +--> DO 10 I=1,100
| |
| |      = WX(I)
| |
| + 10 CONTINUE

```

図 2. 1 2 詳細情報画面(2)--JND2011--

(6) 文単位ベクトライズ情報画面

ループ一覧画面、メッセージ一覧画面、詳細情報画面でSPLIT (PF2) キーを押下すると、プログラム単位ベクトライズ情報画面で選択したルーチンの文単位ベクトライズ情報が、各画面の下側に表示される(図2.13, 図2.14)。図2.14は、図2.13を右側にスクロールして得られる画面である。

この画面では、ベクトライズ/チューニングメッセージ、あるいは詳細情報を参照しながら画面内の原始プログラムの修正ができる(図2.15)。図2.15の最適化制御行(*VOC Lで始まる行)については、続稿[5]を参照されたい。この画面で表示されている情報の内容を以下に示す。

項 目	内 容
行番号	入力した原始プログラムの行番号が表示される。
ベクトルコスト	'nn...n'の形式でベクトライズ表示(V/M/S)の左に表示され、各文をベクトル(ベクトル計算機)で実行したときのおよその実行費用を示す。
ベクトライズ表示	原始プログラムの文の直前に、'V'、'M'、'S'を表示し、その文の並列実行の可否を示す。 V:ベクトル化された。 M:ベクトル化されたものとされないものが混在している。 S:ベクトル化されなかった。 空白:ベクトル化対象外。
文	入力した原始プログラムの文が表示される。
スカラコスト	'nn...nS'の形式で表示され、nn...nによって、各文をスカラ(汎用計算機)で実行したときの実行費用を示す。
実行回数	'nn...nE'の形式で表示され、nn...nによって、各文の実行回数を示す。
TRUE率	'nn, n%'の形式で表示されnn, nによって、スカラ(汎用計算機)で実行したときのIF文の実行回数のうち、'真'となる比率を示す。

```

VECTORIZE - LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600  5920000    71040000    99
00003530-00003530  2983950    35807400    97

EDIT --- F0022.SYSVECT.T1743260.DATA(AFCRVFIT) ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** データの先頭 *****V10L20*****
000001      C
000002      C
000003      C
000004      C
000005      0      SUBROUTINE CRVFIT(AC,A,CA,INEX)
000006                      REAL AC(50,100),A(50,100),CA(50,100)
000007                      COMMON /CM1/N,ML,MT,M,DX/CM7/RF,IEND,IPLT,LIST,KL,KT,P
000008                      *,IOVC,ICAL
000009                      12897 IF(INEX.LT.IOVC) GO TO 37
000010                      15309 IP=MOD(INEX,ICAL)
000011                      14796 IF(IP.NE.0) GO TO 37
000012                      270M DO 20 I=1,N
000013                      155610M CA(I,2)=(1-RF)*CA(I,2)+RF*(-.5*AC(I,1)+.5*AC(I,2))
000014                      3420V DO 20 J=3,M-1

```

図2. 13 文単位ベクトライズ情報画面(1)

```

VECTORIZE - LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600  5920000    71040000    99
00003530-00003530  2983950    35807400    97

EDIT --- F0022.SYSVECT.T1743260.DATA(AFCRVFIT) ----- COLUMNS 073 144
COMMAND ==> SCROLL ==> PAGE
***** データの先頭 *****V10L20*****
000001      |      S      E      00003390
000002      |      S      E      00003400
000003      |      S      E      00003410
000004      |      S      E      00003420
000005      |      OS      OE      00003430
000006      |      S      E      00003440
000007 SI      |      S      E      00003450
000008      |      S      E      00003460
000009      |      12897S      6000E      14.9%      00003470
000010      |      15309S      5103E      00003480
000011      |      14796S      5103E      89.9%      00003490
000012      |      513S      513E      00003500
000013      |      266760S      20520E      00003510
000014      |      20520S      20520E      00003520

```

図2. 14 文単位ベクトライズ情報画面(2)

(7) 原始プログラム修正後のベクトライズ情報

文単位ベクトライズ情報画面で修正したプログラムのベクトライズ情報を得るには、SPLITキーの解除（文単位ベクトライズ情報画面においてENDキー押下）後に、ループ一覧画面またはメッセージ一覧画面において'VCHECK'コマンドを使用する（図2. 16）。

```

VECTORIZE - LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600  5920000    71040000    99
00003530-00003530  2983950    35807400    97

EDIT --- F0022.SYSVECT.T1743260.DATA(AF:CRVFIT) ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** データの先頭 *****V10L20*****
000001          C
000002          C
000003          C
000004          C
000005          0      SUBROUTINE CRVFIT(AC,A,CA,INEX)
000006                  REAL AC(50,100),A(50,100),CA(50,100)
000007                  COMMON /CM1/N,ML,MT,M,DX/CM7/RF,IEND,IPLT,LIST,KL,KT,P
000008                  *,IOVC,ICAL
000009                  *VOCL STMT,IF(15)
000010                  12897 IF(INEX.LT.IOVC) GO TO 37
000011                  15309 IP=MOD(INEX,ICAL)
000012                  *VOCL STMT,IF(90)
000013                  14796 IF(IP.NE.0) GO TO 37
000014                  270M  DO 20 I=1,N

```

図2. 15 文単位ベクトライズ情報画面(3) --プログラム修正例--

```

VECTORIZE - LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> VCHECK_ SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600  5920000    71040000    99
00003530-00003530  2983950    35807400    97
00003580-00003600  520000     720000     40
00003510-00003530  179550     307800     40
*** 終り ***

```

図2. 16 'VCHECK' コマンド入力例

(8) 修正後のプログラムの原始プログラムへの反映

プログラムを十分ベクトル化可能なように修正した後、そのプログラムの内容をもとの原始プログラムに反映させるには、次のいずれかを行えばよい。

- ・ SPLITキーの解除後に、プログラム単位ベクトライズ情報画面、ループ一覧画面、メッセージ一覧画面において 'SAVE' コマンドを入力する (図2. 17)。
- ・ ENDキーを使用して、ベクトライズオプションメニューまで画面を戻す。
- ・ RETURNキーによりプライマリオプションメニューまで画面を戻す。

ただし、INCLUDE文または*INCLUDE行で取り込まれた部分は、もとの原始プログラムには保存されない。

```

VECTORIZE ~ LOOP LIST --- CRVFIT ----- 行 000001 欄 001 080
COMMAND ==> SAVE_ SCROLL ==> PAGE

LINE-NUMBER      V-COST      S-COST      V-LENG
00003600-00003600 5920000    71040000    99
00003530-00003530 2983950    35807400    97
00003580-00003600 520000     720000     40
00003510-00003530 179550     307800     40
*** 終り ***

```

図2. 17 'SAVE' コマンド入力例

3. Fortranプログラム実行解析ツールFORTUNE

3. 1 機能

ソースプログラムの動的な実行の流れを追跡して、どの部分に実行が集中しているかを知るためには、FORTUNE [3] を使用するとよい。FORTUNEの出力を、図3. 1に示す。プログラム単位ごとに以下のような情報が出力される。

- ・実行文それぞれの実行回数と実行費用
- ・プログラム単位ごとの実行回数と実行費用
- ・論理IF文、ブロックIF文、ELSE IF文において論理式が真となる場合の実行回数とその比率（真率）

これらの情報は、前述のように、パッチ型ベクトライザで動的効果表示（FEATURE=D）を指定することにより、ベクトライズ情報とともに出力することができるので、そちらの方を勧める。

ただし、会話型ベクトライザでFORTUNEモードを選択する場合や、ベクトライズモードでFORTUNEの実行結果を使用して動的なベクトライズ情報を得たい場合などは、あらかじめTSSでFORTUNEコマンドを実行し、FORTUNE実行解析データセットを出力しておく必要がある。

FORTUNE V10L10			DATE 85.09.20 TIME 09.45.19		PAGE 1
=EXECUTIONS=	=TRUE=	=ISN=	=STATEMENT=	=SEQ=	=COST=
		C	*****MAIN PROGRAM*****	00000300	
		00001	REAL X(100),Y(50,100),DY(50),YD(50,50),CS(50,50),DTA(50),D(50,100)	00000400	
			1,U(50,100),V(50,100),P(50,100),DC(50,100),UC(50,100),VC(50,100)	00000500	
			1,PC(50,100)	00000600	
			1,AA(50,100),AB(50,100),AC(50,100),BA(50,100),BB(50,100),BC(50,100)	00000700	
			1,CD(50,100),CP(50,100),CU(50,100),CV(50,100),M1,M2	00000800	
			1,ZM(50,100),ZK(50,100)	00000900	
		00002	COMMON/CM1/N,ML,MT,M,DX/CM2/M1,M2,RATIO/CM3/DY,YD,CS,DTA,DTA1,DY13	00001000	
			1,CS1,CS3,YD1,YD3/CM5/D,P,U,V/CM6/DC,PC,UC,VC	00001100	
			1/CM7/RF,IEND,IPLT,LIST,KL,KT,PS1,IOVC,ICAL/CM4/BL,T,BET1,BET2,	00001200	
			1RAMDA/CM8/X,Y/CM9/CD,CP,CU,CV/CM10/XP,YP,XMAX,YMAX	00001300	
		C		00001400	
1		00003	INEX=0	00001500	1
1		00004	P2=1/RATIO/7	00001600	15
1		00005	DO 30 J=1,M	00001700	1
100		00006	DO 30 I=1,N	00001800	100
4000		00007	CD(I,J)=0	00001900	4000
4000		00008	CP(I,J)=0	00002000	4000
4000		00009	CU(I,J)=0	00002100	4000
4000		00010	30 CV(I,J)=0	00002200	4000
		C		00002300	
1		00011	CALL ORTH	00002400	1
		CF	CALL PLOTS(0.0,0.0,16)	00002500	
		CF	CALL FACTOR(0.8)	00002600	
1		00012	CALL SCTR	00002700	1
		C		00002800	
1		00013	CALL INICON	00002900	1
		C		00003000	
2000		00014	130 INEX=INEX+1	00003200	4000
		C		00003300	
2000		00015	DO 20 I=1,N	00003400	2000
80000		00016	DO 10 J=1,M-1	00003500	80000
7920000		00017	AA(I,J)=D(I,J)*U(I,J)	00003600	23760000
7920000		00018	AB(I,J)=DC(I,J)*UC(I,J)	00003700	23760000
7920000		00019	10 AC(I,J)=DC(I,J)*VC(I,J)	00003800	23760000
80000		00020	20 AA(I,M)=D(I,M)*U(I,M)	00003900	240000
2000		00021	CALL NEWSTP(AA,AB,AC,DC)	00004000	2000
2000		00022	CALL SMOOTH(DC)	00004100	2000
		C		00004200	
2000		00023	CALL CRVFIT(DC,D,CD,INEX)	00004300	2000
		C		00004400	
		C		00004500	
2000		00024	DO 40 J=1,M-1	00004600	2000
198000		00025	DO 40 I=1,N	00004700	198000
7920000		00026	40 PC(I,J)=DC(I,J)=(1-(UC(I,J)*=2+VC(I,J)*=2))/7	00004800	253440000
2000		00027	CALL SMOOTH(PC)	00004900	2000
2000	299(14.9%)	00028	IF(INEX.LT,IOVC) GO TO 37	00005000	4299
1701		00029	IP=MOD(INEX,ICAL)	00005100	5103
1701	1530(89.9%)	00030	IF(IP.NE.0) GO TO 37	00005200	4932
		C		00005300	
171		00031	DO 50 I=1,N	00005400	171
6840		00032	CP(I,1)=(1-RF)*CP(I,1)+RF*(0.5*PC(I,1)-0.5*PC(I,2))	00005500	82080
6840		00033	CP(I,M-1)=(1-RF)*CP(I,M-1)+RF*(-.5*PC(I,M-1)+.5*PC(I,M-2))	00005600	116280
6840		00034	DO 50 J=2,M-2	00005700	6840
663480		00035	50 CP(I,J)=(1-RF)*CP(I,J)+RF*(.375*PC(I,J-1)-.25*PC(I,J)-	00005800	11279160

Σ 中略

FORTUNE V10L10			DATE 85.09.20 TIME 09.45.19		PAGE 15
=ROUTINE=	=EXECUTIONS=	=COST=	%	0.....1.....2.....3.....	
*MAIN	1	649761450	32.8	I*****	
ORTH	1	135108	0.0	I	
INICON	1	179855	0.0	I	
NEWSTP	4000	638820000	32.3	I*****	
CRVFIT	6000	108064410	5.5	I*****	
SMOOTH	8000	581592000	29.4	I*****	
OUTPUT	20	1495540	0.1	I	
LISTUP	1	2322	0.0	I	
FLCON	1	55566	0.0	I	
SCTR	1	20620	0.0	I	
1980326891			=== SUM TOTAL FOR THIS PROGRAM ===		

図3. 1 FORTUNEによる実行解析情報

3. 2 FORTUNEコマンドの使用法

コマンド	オペランド
FORTUNE	データセット名 'コンパイラオプション' [LIST (* データセット名) SYSLIST (出力クラス)]

・オペランドの説明

データセット名：解析したいソースプログラムの格納されているデータセット名を指定する。

'コンパイラオプション'：Fortran 77 コンパイラオプションを指定する。ソースプログラムの形式は、省略値が自由形式になっているので、標準形式の場合には'FIX'と指定する必要がある。

LIST (* | データセット名)：FORTUNE実行解析結果の出力先を指定する。*と指定すれば端末に出力される。省略時は、データセット出力（解析情報データセット）である。

SYSLIST (出力クラス)：FORTUNE実行解析結果をラインプリンタに出力する時の出力クラスを指定する。

FORTUNEコマンドを実行すると、自動的に次の3つのデータセットが生成されるので、会話型ベクトライザの入力データとして使用するのではなく、消去して構わない。

- ・データセット名.LIST ・・・ FORTUNE実行解析データセット。会話型ベクトライザで動的解析を行う場合、必要となる。
- ・データセット名.SYSINF ・・・ FORTUNE編集情報データセット
- ・データセット名.SYSFIN ・・・ FORTUNE編集済みソースプログラムデータセット

4. FORDAP 77

FORTUNEと同様の情報を出力するシステムにFORDAP 77がある。FORDAP 77を使用することでも、プログラムのどの部分が実行の中心かを知り、チューニングのターゲットとすることができる。詳しい使用法は[4]に譲るとして、ここでは簡単なコマンド入力形式を紹介するとともに、図4. 1に同じテストプログラムをバッチ型FORDAP 77にかけて得た結果を掲げるので、参考にされたい。

FACOM OSIV/F4 FORDAP77 -840225- V-03 L-01 DATE 85-09-20 TIME 09-14-35

***** SUMMARY OF PROCEDURES IN SOURCE PROGRAM *****

NAME	NO	LINE	EXECUTABLE	UNEXECUTED	INVOKED	EXECUTIONS	TIME
M *****	1	168	122	0	0	113517936	161418
S CRVFIT	6	29	19	0	6000	26330712	15400
S FLCON	10	28	16	0	1	395	8
S INICON	4	46	33	0	1	32370	9
S LISTUP	9	17	9	0	1	152	1
S NEWSTP	5	66	45	0	6000	96864000	41560
S ORTH	3	43	35	0	1	6931	21
S OUTPUT	8	46	31	0	20	10840	225
S SCTR	11	20	13	0	1	8307	2
S SMOOTH	7	36	22	0	8000	63720000	42070
D *****	2	15	1	0	0	0	0
*****TOTAL*****		514	346	0	20025	300491623	

SOURCE STATEMENT	EXECUTION TIME	161418 MSEC	EXECUTIONS	TRUE
C *****MAIN PROGRAM *****			00000300	
REAL X(100),Y(50,100),DY(50),YD(50,50),CS(50,50),DTA(50),D(50,100)			00000400	
1,UC(50,100),VC(50,100),P(50,100),DC(50,100),UC(50,100),VC(50,100)			00000500	
1,PC(50,100)			00000600	
1,AA(50,100),AB(50,100),AC(50,100),BA(50,100),BB(50,100),BC(50,100)			00000700	
1,CD(50,100),CP(50,100),CU(50,100),CV(50,100),M1,M2			00000800	
1,ZM(50,100),ZK(50,100)			00000900	
COMMON/CM1/M,ML,MT,M,DX/CM2/M1,M2,RATIO/CM3/DY,YD,CS,DTA,DTA1,DY1300001000			00001100	
1,CS1,CS3,YD1,YD3/CM5/D,P,U,V/CM6/DC,PC,UC,VC			00001200	
1/CM7/RF,IEND,IPLT,LIST,KL,KT,PSI,IOVC,ICAL/CM4/BL,T,BET1,BET2,			00001300	
1RAMDA/CM8/X,Y/CM9/CD,CP,CU,CV/CM10/XP,YP,XMAX,YMAX			00001400	
C INEX=0			00001500	1
P2=1/RATIO/7			00001600	1
DO 30 J=1,M			00001700	1
DO 30 I=1,N			00001800	100
CD(I,J)=0			00001900	4000
CP(I,J)=0			00002000	4000
CU(I,J)=0			00002100	4000
30 CV(I,J)=0			00002200	4000
C CALL ORTH			00002300	1
CF CALL PLOTS(0.0,0.0,16)			00002400	
CF CALL FACTOR(0.8)			00002500	
CALL SCTR			00002600	
C CALL INICON			00002700	1
C CALL IMICON			00002800	
C 130 INEX=INEX+1			00002900	1
C DO 20 I=1,N			00003000	
DO 20 J=1,M-1			00003200	2000
AA(I,J)=D(I,J)+U(I,J)			00003300	
AB(I,J)=DC(I,J)+UC(I,J)			00003400	2000
10 AC(I,J)=DC(I,J)+VC(I,J)			00003500	80000
20 AA(I,M)=D(I,M)+U(I,M)			00003600	7920000
CALL NEWSTP(AA,AB,AC,DC)			00003700	7920000
CALL SMOOTH(DC)			00003800	7920000
C CALL CRVFIT(DC,D,CD,INEX)			00003900	80000
C DO 40 J=1,M-1			00004000	2000
DO 40 I=1,N			00004100	2000
40 PC(I,J)=DC(I,J)+((1-(UC(I,J)+VC(I,J)+2))/7			00004200	
CALL SMOOTH(PC)			00004300	2000
IF(INEX.LT.IOVC) GO TO 37			00004400	
IP=MOD(INEX,ICAL)			00004500	
IF(IP.NE.0) GO TO 37			00004600	
C DO 50 I=1,N			00004700	2000
CP(I,1)=(1-RF)*CP(I,1)+RF*(0.5*PC(I,1)-0.5*PC(I,2))			00004800	198000
CP(I,M-1)=(1-RF)*CP(I,M-1)+RF*(-.5*PC(I,M-1)+.5*PC(I,M-2))			00004900	7920000
DO 50 J=2,M-2			00005000	2000
50 CP(I,J)=(1-RF)*CP(I,J)+RF*(.375*PC(I,J-1)-.25*PC(I,J)-			00005100	2000
1.125*PC(I,J+1))			00005200	1701
37 CONTINUE			00005300	1701
			00005400	171
			00005500	6840
			00005600	6840
			00005700	6840
			00005800	663480
			00005900	
			00006000	2000

図4. 1 FORDAP77による実行解析情報

コマンド	オペランド
FORDAP77	データセット名 [OBJ (オブジェクトモジュール名)] [LIB (ロードモジュール名)] [DATA (データセット名)] [RESULT (データセット名)] [SY (出力クラス)] [TYPE (N W)] [TERMINAL (端末出力行数)]

・オペランドの説明

データセット名：ソースプログラムが格納されているデータセット名を指定する。

OBJ (オブジェクトモジュール名)：ソースプログラムとコンパイル済みのオブジェクトモジュールを結合して実行する時、そのオブジェクトモジュール名を指定する。

LIB (ロードモジュール名)：ソースプログラムと作成済みのロードモジュールを結合して実行する時、そのロードモジュール名を指定する。

DATA (データセット名)：入力データが格納されているデータセット名を指定する。省略時は、端末入力となる。

RESULT (データセット名)：実行結果を格納したいデータセット名を指定する。省略時は実行結果を端末に出力する。

SY (出力クラス)：出力クラスを指定する。省略時は端末に出力する。

A：PL/I 60 字セット LP
 O：日本語及び図形出力用 LP (NLP)
 S：英小文字付き LP (NLP)
 K：カナ文字付き LP (NLP)
 U：オープン NLP

TYPE (N | W)：出力形式を指定する。

N：出力欄 80 桁を想定した出力形式 (TSS 向き)
 W：出力欄 132 桁を想定した出力形式 (BATCH 向き)

TERMINAL (端末出力行数)：端末出力を指定行数ごとに一時停止させる時指定する。

参考文献

1. 磯辺, 松延, 今村, 武富, 景川 ベクトルプロセッサFACOM VP100について, 九州大学大型計算機センター広報, 18, 5, 1985, 397-412.
2. 計算機マニュアル FACOM OSIV/F4 MSP 会話型ベクトライザ使用手引書 (78SP-5690-2), 富士通 (株)
3. 計算機マニュアル FACOM OSIV/F4 MSP FORTUNE使用手引書 (78SP-5360-1), 富士通 (株)
4. 牛島, 程 Fortran77プログラム動的解析ツール-Fordap77の使用について-, 九州大学大型計算機センター広報, 17, 4-5, 1984, 276-287.
5. 武富, 桜井, 平野, 景川 ベクトルプロセッサの有効な使用法 (2) -プログラムの高速化について-, 九州大学大型計算機センター広報, 18, 6, 1985.

付録 1. タイマルーチンによる計算時間の測定

測定したい各サブルーチンなどに、次のようにサービスサブルーチンCLOCKを挿入する。

・最初の実行文の前

```
CALL  CLOCK(IT1,2,0)
```

```
CALL  CLOCK(IT2,2,0)          . . . . . CLOCK呼び出しのオーバーヘッド
```

・RETURN文の前

```
CALL  CLOCK(IT3,2,0)
```

```
IVCPU (n, 1) = IVCPU (n, 1) + (IT3-IT2) - (IT2-IT1)
```

```
. . . . . CPU時間の集計
```

```
IVCPU (n, 2) = IVCPU (n, 2) + 1
```

```
. . . . . 呼び出し回数の集計
```

ここで、nは測定対象ルーチンのID番号とする。

注) サービスサブルーチンCLOCKについて

```
CALL  CLOCK (g [, i1 [, i2]])
```

実行開始からのCPU時間がi1に示す単位でgに返される。返される値のデータの型をi2で指定する。

i1 = 0	... <u>秒単位</u>	i2 = 0	... <u>4バイトの整数値</u>
= 1	... ミリ秒単位	= 1	... 4バイトの実数値
= 2	... マイクロ秒単位	= 2	... 8バイトの実数値
		= 3	... 16バイトの実数値

(下線は、省略値である)